

## 浪脚本零起点入门系列(十一)

### 样条线编辑 Editable Spline

作者：飞浪

声明:本教程为 **CG++** 原创,转载请注明出处,谢谢合作:)

本节关键词: SplineShape

-----我是废话的分割线-----

当我点开飞浪脚本零起点入门系列第一节时,发现发表日期为 2008 年 9 月 2 号。到如今已近两年了。两年,足以恋个爱生个娃了。不知道两年前跟我们一起学脚本的人如今是否还在坚持...

-----我是正题的分割线-----

线条塌陷后,用 classof \$检查会发现有两种类型, SplineShape 和 line, line 是由线条创建面板里面的 line 创建的,塌陷后还是 line。这里,本人主要讲 SplineShape。这节里,我会讲如何创建样条线以及如何利用样条线工作。后面用到的样条线函数,在帮助里面搜索 SplineShape 和 shape 都可以找到。

创建样条线

创建一条线段:

```
ss=SplineShape ()  
addNewSpline ss  
addKnot ss 1 #corner #line [0,0,0]  
addKnot ss 1 #corner #line [100,0,0]  
updateshape ss
```

运行以上所有代码,就创建了一条线段,线段一个点在[0, 0, 0],另一个点在[100, 0, 0]。以上代码包涵了创建样条线的最基本流程,下面逐步解析:

ss=SplineShape ()

--首先创建一个 spline 并赋给变量 ss,注意,此时此线条是空的,什么都没有。

addNewSpline ss

--然后在此线条里面加一条线 spline,此时线条还是空的,因为没有点。

addKnot ss 1 #corner #line [0,0,0]

addKnot ss 1 #corner #line [100,0,0]

--再在线条里面加点,1表示在第一个 spline 里面加点,#corner 表示角点,有#smooth | #corner | --#bezier | #bezierCorner 可供选择。#line 表示是直线,有#curve | #line 可供选择。

updateshape ss

—最后更新线条，注意这一步必须要，否则 MAX 会挂的。

创建一个矩形：

```
ss=SplineShape ()
addNewSpline ss
addKnot ss 1 #corner #line [0,0,0]
addKnot ss 1 #corner #line [100,0,0]
addKnot ss 1 #corner #line [100,100,0]
addKnot ss 1 #corner #line [0,100,0]
close ss 1
updateshape ss
```

观察上面的代码，首先把矩形的四个点加入到线条中，这样只会形成一个 U 字形的线框，最后用 close 关闭线条就可以。

画多条线：

```
ss=SplineShape ()
addNewSpline ss
addKnot ss 1 #corner #line [0,0,0]
addKnot ss 1 #corner #line [100,0,0]
addNewSpline ss
addKnot ss 2 #corner #line [0,20,0]
addKnot ss 2 #corner #line [100,20,0]
addNewSpline ss
addKnot ss 3 #corner #line [0,40,0]
addKnot ss 3 #corner #line [100,40,0]
updateshape ss
```

上面一共用了 3 次 addNewSpline，画了 3 条线

合并两个线条：

```
a=Rectangle()
b=Circle()
convertToSplineShape a
convertToSplineShape b
addAndWeld a b 0
```

创建一个矩形和一个圆塌陷成样条线，将 b 合并到 a 里面。addAndWeld 最后的 0 表示焊接的阈值，如果合并后 a, b 线条的终点距离在此范围之内就将焊接到一起。

关于 splineshape 的函数请看这里：

[http://www.cgplusplus.com/online-reference/maxscript-reference/source/splineshape\\_shape.htm](http://www.cgplusplus.com/online-reference/maxscript-reference/source/splineshape_shape.htm)

沿路径复制：

画一条曲线并保持选中，运行以下脚本：

```
for i in 0 to 1 by 0.1 do
(
p=pathInterp $ 1 i
sphere pos:p
)
```

可以看到沿着此线条创建了球体

pathInterp 路径插值，根据路径控制的比例（基于点），用 0.0 到 1.0 之间数值返回线条上的一个对应点坐标。

对线条的一些操作方法：

[http://www.cgplusplus.com/online-reference/maxscript-reference/source/shape\\_common\\_properties\\_operators\\_and\\_methods.htm](http://www.cgplusplus.com/online-reference/maxscript-reference/source/shape_common_properties_operators_and_methods.htm)

下面这个实例是分离样条线里面的线条，里面对一些常用函数作了说明：

```
fn fl_detachSplines ss =
( --分离线条,By feilang ,from CG++

numsp = numsplines ss --获取线条 spline 数量

sp=#()--用于收集分离后的线条

if numsp > 1 then
( --遍历每个 spline，然后记录每一个点的信息，根据这些信息重新生成线条

for i in 1 to numsp do

( tempS = splineShape pos:(getKnotPoint ss i 1) --创建一个空样条线

addNewSpline tempS --增加一个线条 spline

for k in 1 to numKnots ss i do --遍历原样条线中某个 spline 上的所有点，并增加到新创建的线条中

--addKnot 是加点，getKnotType 获取原线条点的点类型，getKnotPoint 获取原线条点的坐标

--getInVec 和 getOutVec 是获取原线条点的手柄坐标，这些信息都添加到对应的新点上面
```

```
addKnot tempS 1 (getKnotType ss i k) #curve (getKnotPoint ss i k) (getInVec ss i k)
(getOutVec ss i k)
if isclosed ss i then close tempS 1 --判断原线条是否关闭, 如果是关闭的, 新线条也要关闭
updateShape tempS --更新线条, 必不可少!!!
append sp tempS
)
)
delete ss --删除原线条
sp
)
```

运行后, 直接使用 `fl_detachSplines $`

下面把分离后的线条选中, 运行代码:

```
fn fl_collapseSplines ssarr =
( --合并线条, By feilang, from CG++
ss=for s in ssarr where classof s == splineshape or classof s == line collect s
for i in 2 to ss.count do addAndWeld ss[1] ss i
if ss.count!=0 then updateshape ss[1]
ss[1]
)
```

再运行 `fl_collapseSplines $`, 没错, 它们又合并到一起了。

认真理解上述代码, 你对 spline 的操作概念应该会有个了解, 其实可以把 splineshape 简单的看作一个多维数组, splineshape 下面有 spline1, spline2, spline3... 每个 spline 下面又有点 1, 2, 3... 所以要知道一个点 knot, 就要知道它是属于哪个 spline。更多实用的函数躺在帮助文档里面, 期待你去唤醒它们。

本节原帖地址: <http://www.cgplusplus.com/bbs/viewthread.php?tid=1443>

本节结束。